

Using “AsSeenBy” In Alice

By Jenna Hayes
under the direction of Professor Susan Rodger
Duke University
July 2008



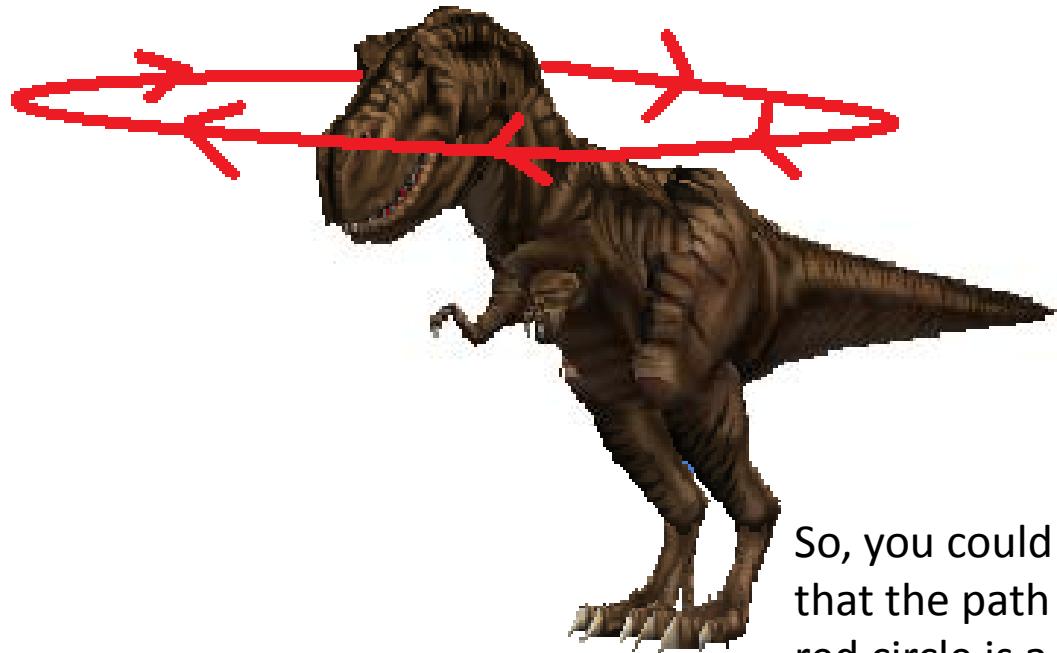
Step 1: Understanding AsSeenBy

Download the Alice world that goes along with this tutorial. Also download the “jaws” wav file, and save it to your desktop. Open up the world in Alice.

AsSeenBy is a condition in Alice that changes the way an object carries out a method. Lets look at an example to explain it fully.

We’ll use the t-rex and the shark in the Alice world to demonstrate AsSeenBy.

Put the turn method in your method editor to tell the t-rex to turn right one revolution. Play your world, and visualize the path of the t-rex’s eyes as he turns. It would look something like this:



So, you could say that the path of the red circle is a right turn *as it is seen by* a t-rex.

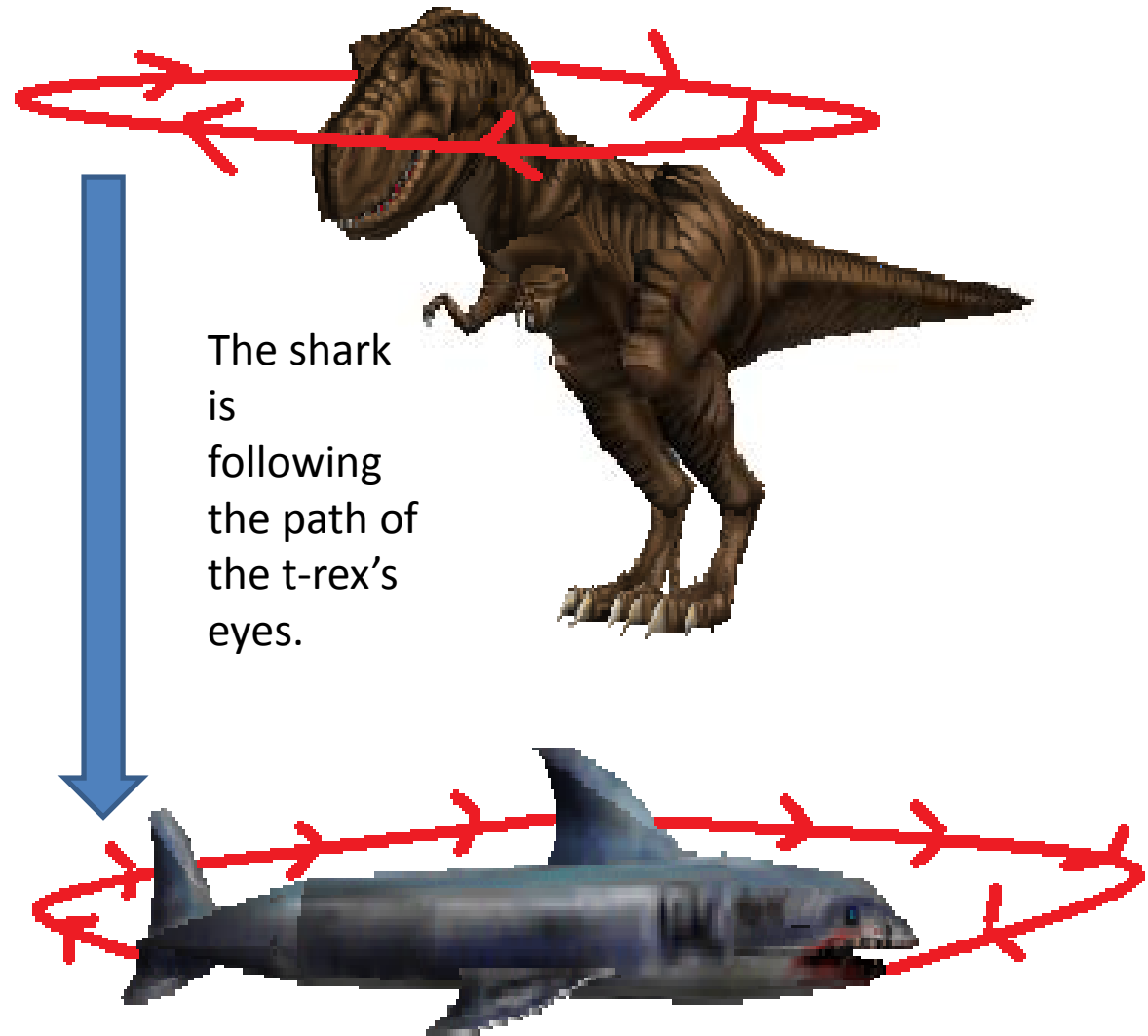
Step 1 cont.



Delete the command for the t-rex to turn right. Now add a command for the shark to turn right 1 revolution. This time, click on **more...** next to the command, and click on **asSeenBy**. Then click on **t-rex**, and **the entire t-rex**.

Step 1 cont.

Play your world. You'll notice that instead of turning right normally, the shark zooms in a circle around the t-rex. Think of it this way: the shark is following the path that a t-rex's eyes would make if *it* were making a right turn. So the shark is turning right as seen by the t-rex.



Step 2: Rolling with AsSeenBy

Now we'll use AsSeenBy to complete the Alice world. Delete the command for the shark to turn right, and in the events editor change **When the world starts** to **world.sharkask**. There are two possible events in this world. When the world is run, the shark asks for a coconut. If you press Y, the shark gets a coconut. If you press N, nothing happens yet, but we will add a method there later.

Play the world and press Y to see what the **trexGiveCoconut** method does.



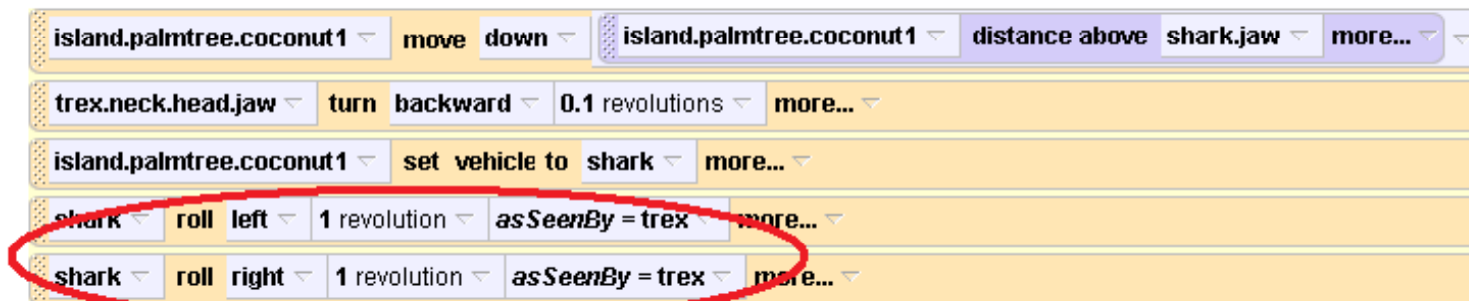
Step 2 cont.

After the shark gets the coconut, we want to make him celebrate. Open up the **trexGiveCoconut** method so you can add commands to the end of the method to make the shark celebrate. Using **asSeenBy**, we're going to make the shark do two flips in the air. Use the **roll** method, and tell the shark to roll left one revolution **asSeenBy** the t-rex, and then right one revolution **asSeenBy** the t-rex. Your code will look like this:

Play your world and press Y to see what the shark's rolls look like. He does two flips in the air, following the path the t-rex's eyes would follow if he were to roll.

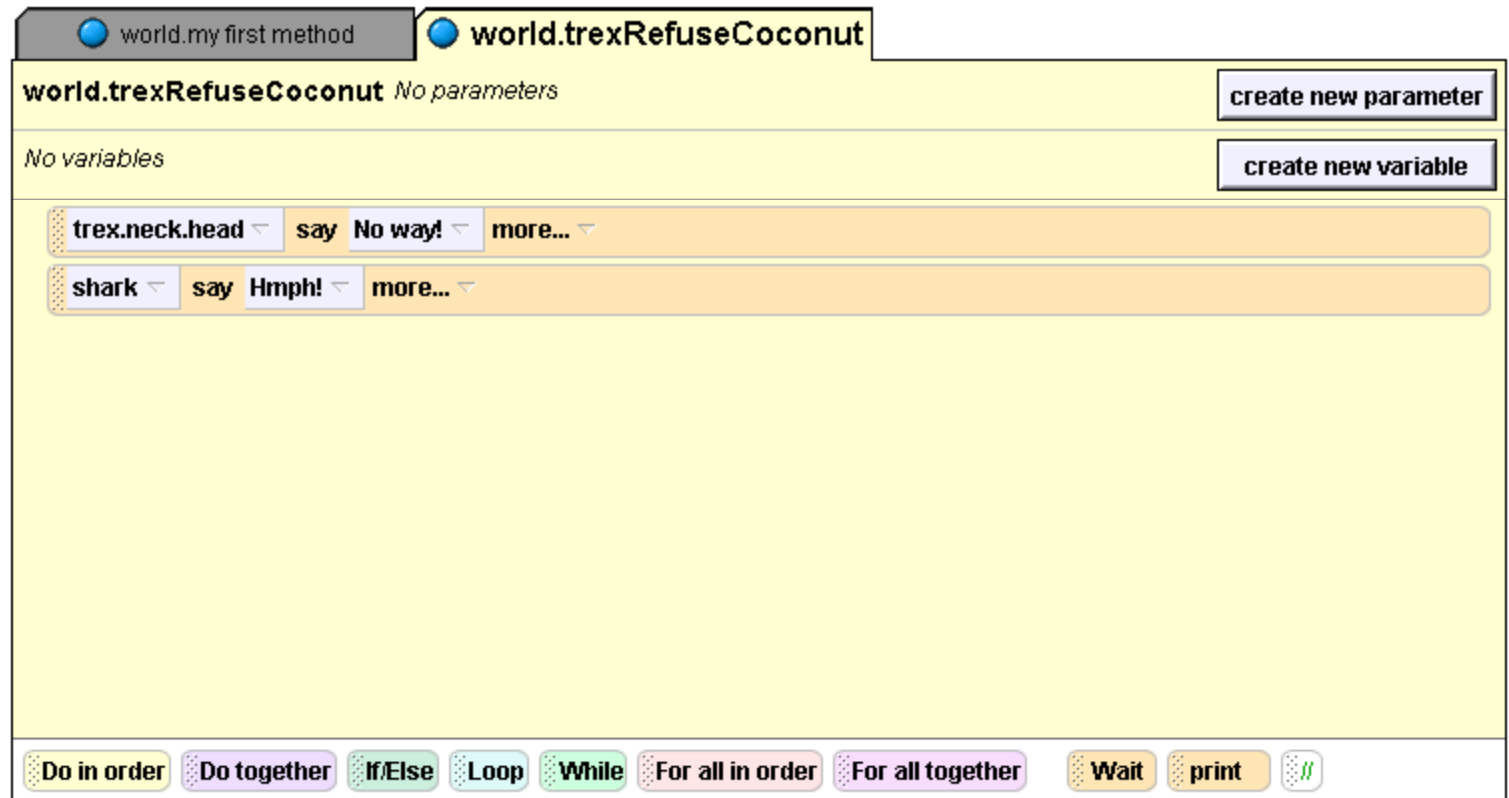


If you want, add a **Do Together** and tell the shark to say something like "Thanks!" or "Whoopee!" while flipping in the air.



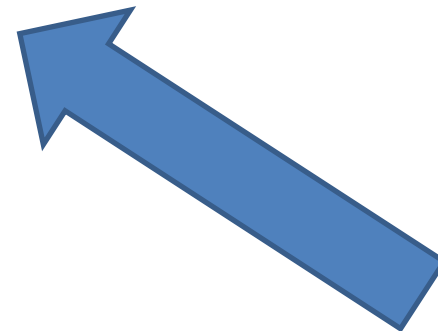
Step 3: Refusing the Coconut

Now we need to tell the objects what to do when you press N on the keyboard, and the shark doesn't get a coconut. Create a new world method called **trexRefuseCoconut**. Place this method in the events editor next to **When N is typed**.



The image shows the Scratch events editor for a world method named **world.trexRefuseCoconut**. The editor has two tabs: **world.my first method** and **world.trexRefuseCoconut**. The **world.trexRefuseCoconut** tab is active, showing **No parameters** and **No variables**. There are buttons for **create new parameter** and **create new variable**. The main area contains two code blocks: **trex.neck.head** **say** **No way!** **more...** and **shark** **say** **Hmph!** **more...**. At the bottom, there is a palette of control blocks: **Do in order**, **Do together**, **If/Else**, **Loop**, **While**, **For all in order**, **For all together**, **Wait**, **print**, and a **//** block.

The first thing will be to tell the t-rex to say “No way!”. Then tell the shark to say “Hmph!”
Your code will look like this:



Step 3: cont.

Now to use `AsSeenBy`. We want the shark to circle the island menacingly after he says “Hmph!”. We could tell him to turn left `asSeenBy` the t-rex, but when that is commanded he swims *through* part of the island. Instead, lets make him turn left 1 revolution `asSeenBy` the *island*. Create a new world method called **sharkCircle**. Tell the shark to turn left 1 revolution `asSeenBy` the island.



Step 4: Finishing Up

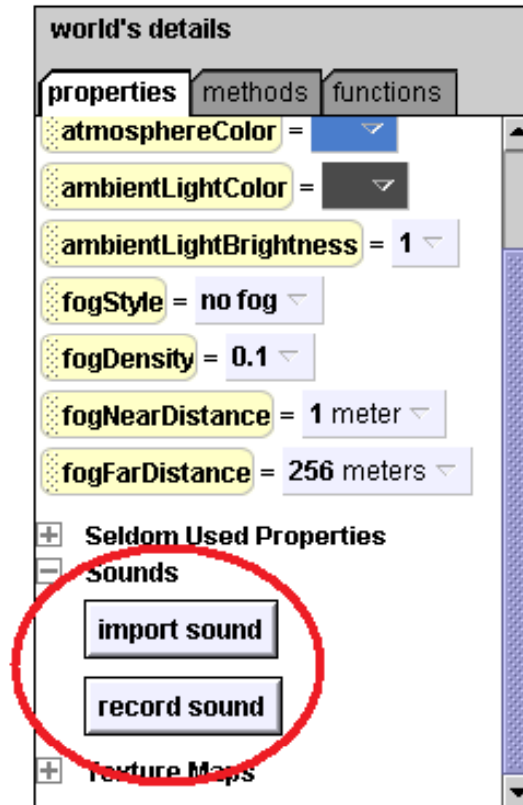
Drag and drop a call to `world.sharkCircle` method in your `world.trexRefuseCoconut` method. Play your world and press N to watch it. The shark looks a little too fast when it turns around the island, so to make it more menacing, we'll slow it down. Change the duration of the left turn from 1 second to 2 seconds. Play your world and press N to see the difference.

shark ▾ turn left ▾ 1 revolution ▾ asSeenBy = island ▾ duration = 2 seconds ▾ more... ▾

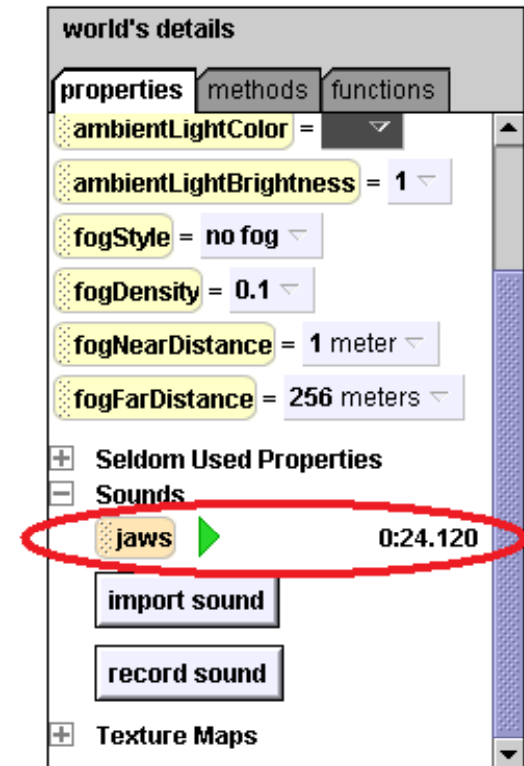


Step 5: Adding a Sound

To make our world complete, we're going to add the jaws theme, so that it plays while the shark is circling the island. Click on the world in your object tree, and then go to the **properties** tab. Click the plus sign next to the word **Sounds**.



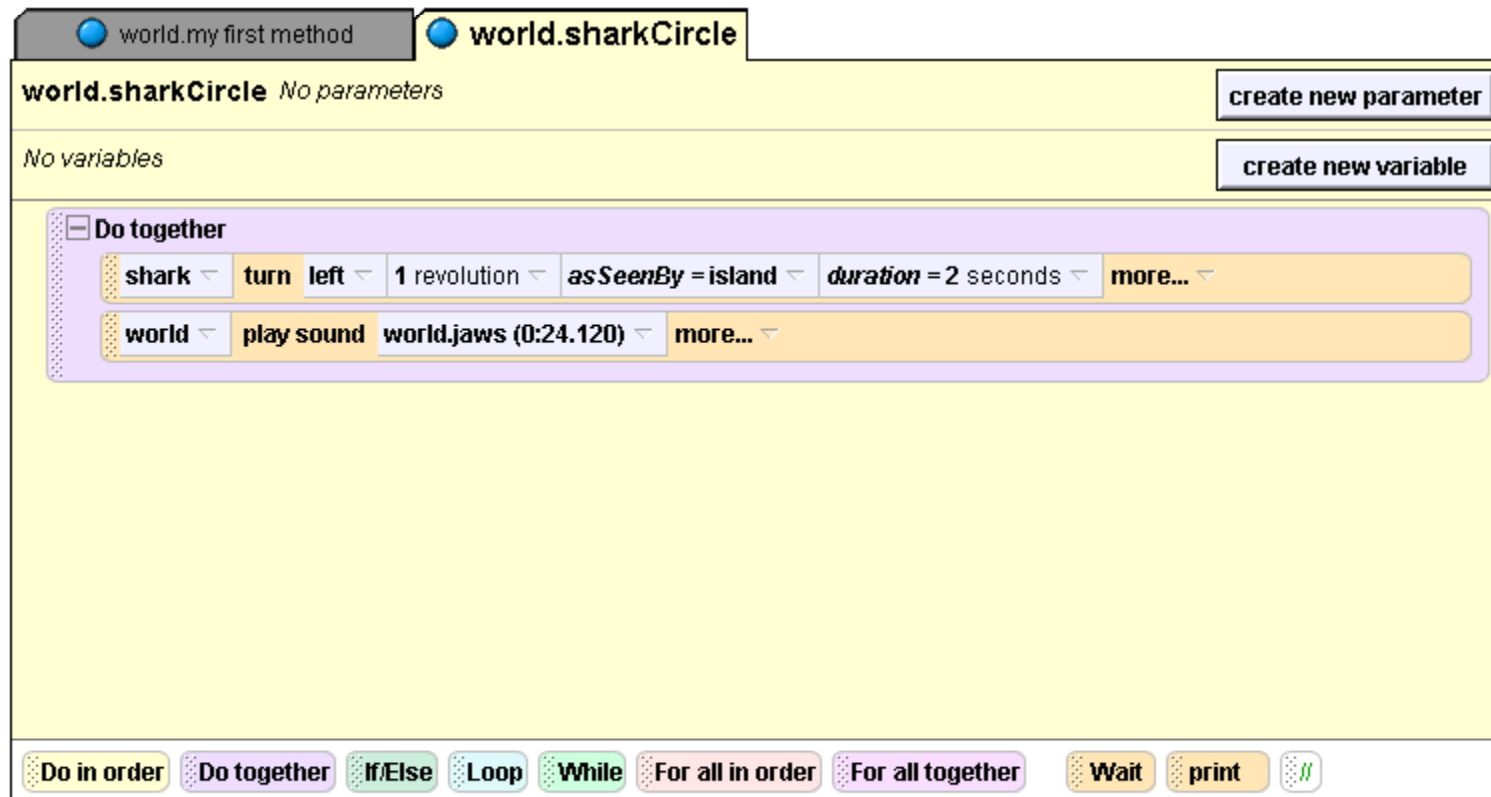
Now click on **import sound**, and find the jaws wav that you saved on your desktop. double-click it, and it should appear there under sounds. You can see the name of the file, and its length.



You can click on the word **jaws** to hear the sound.

Step 5: cont.

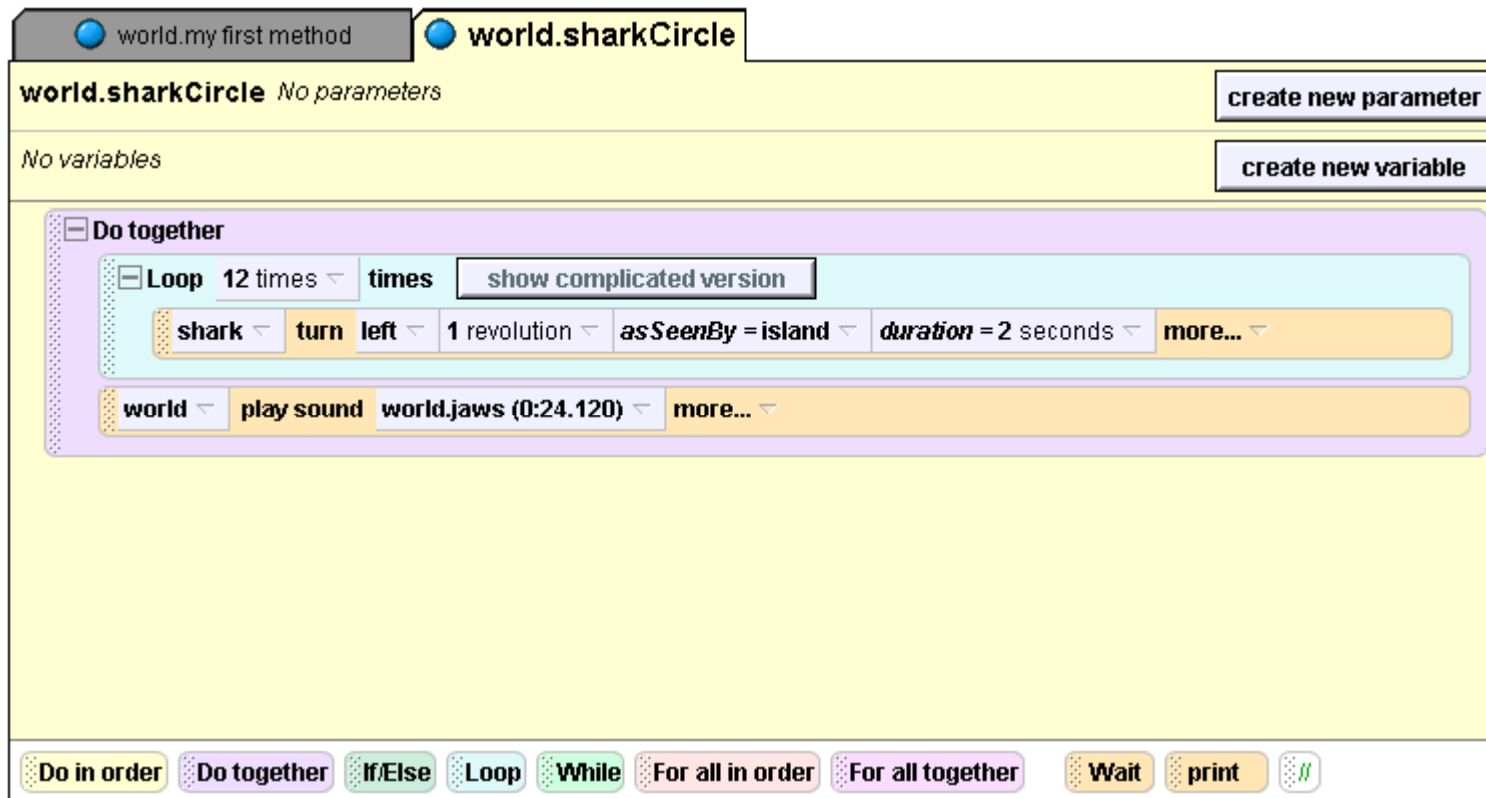
Go back to your `world.sharkCircle` method. Add a Do Together, and put your left turn command inside it. Then drag the `jaws` sound into the Do Together. Your code will look like this:



Play your world and press N to watch it. You'll notice that the song keeps playing long after the shark finishes circling. Let's fix that.

Step 6: Adding a Loop

To make the shark keep circling until the music is finished, let's put the **turn left** command in a loop. If each turn takes **2 seconds**, and the sound file lasts **24.12 seconds**, we should make the loop run **12 times**. Your code will look like this:



Play the world and press N to see what it looks like.

Step 6: cont.

You can see that the shark circles two extra times after the music stops playing. This could be because the last two seconds of the sound file are just silence. To make it look better, change the number of times it loops from **12** to **10**. Play the world and press N to see if it looks better.

The image shows a Scratch code editor window with two tabs: 'world.my first method' and 'world.sharkCircle'. The 'world.sharkCircle' tab is active, showing a script with the following structure:

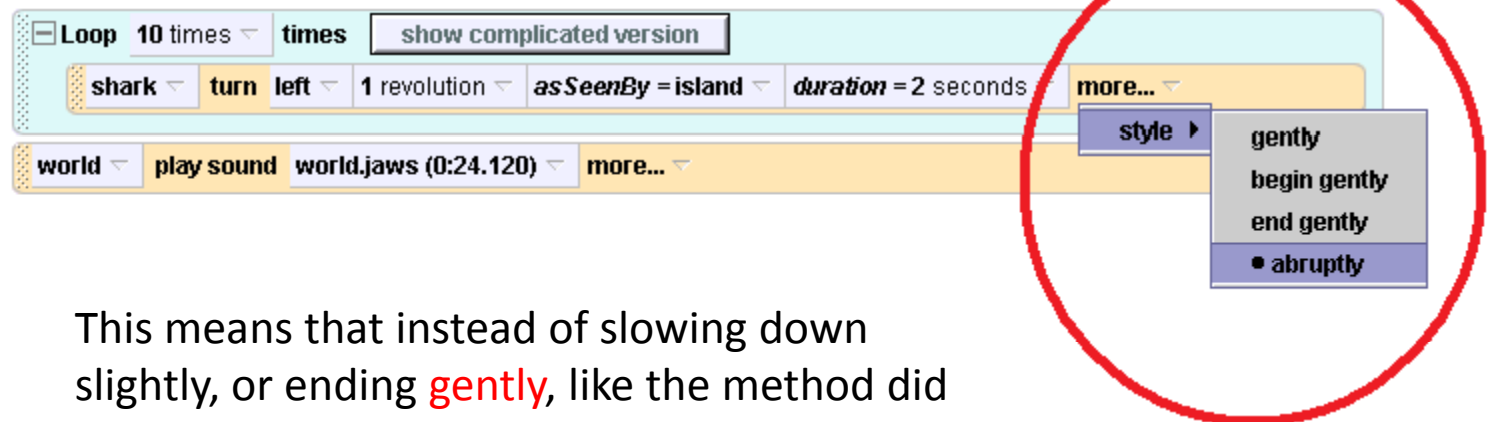
- Script name: **world.sharkCircle** (No parameters)
- Variables: **No variables**
- Block type: **Do together** (purple block)
- Inside 'Do together':
 - Loop** block (light blue) with '10 times' and 'times' dropdown, and a 'show complicated version' button.
 - shark** block (orange) with 'turn left' and '1 revolution' dropdown, and 'asSeenBy = island' dropdown.
 - world** block (orange) with 'play sound' and 'world.jaws (0:24.120)' dropdown.

At the bottom of the editor, there is a palette of block types: Do in order, Do together, If/Else, Loop, While, For all in order, For all together, Wait, print, and a comment block.

Step 7: The Last Touch

You should see that between each time that the shark circles the island, he slows down and almost stops. We want him to circle the island smoothly until he is finished. To do this, we will change the **style** of the turn command.

On the line in your method editor that tells the shark to turn, click on **more...** and select **style**. You should then select **abruptly**:



This means that instead of slowing down slightly, or ending **gently**, like the method did when the shark was circling, it will simply stop and start without slowing down first. Play your world and press N to make sure it looks better.

Congratulations! Now your shark can be happy if he gets a coconut, and angry if he doesn't get one.

